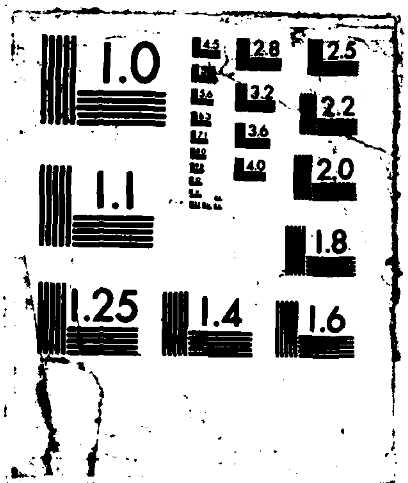
AD-A189 134 OCEANOGRAPHIC EXPORT SYSTEM DEVELOPMENT-YEAR 1(U) 1/1
PERCEPTICS CORP KNOXVILLE TN M G THOMASON NOV 87
N00014-87-C-6001

UNCLASSIFIED F/G 8/3 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

(2)

Oceanographic Expert System Development

Final Report - Year 1

[illegible] N [illegible] 7-[illegible]001

Prepared for:
Naval Ocean Research and Development Activity
NSTL, Mississippi 39529-5004

November, 1987

perceptics

2

Oceanographic Expert System Development
Final Report - Year 1

Contract No. N00014-87-C-6001

Prepared for:
Naval Ocean Research and Development Activity
NSTL, Mississippi 39529-5004

November, 1987

Oceanographic Expert System Development
Final Report - Year 1

Contract No. N00014-87-C-6001

Prepared by:

M. G. Thomason

Perceptics Corporation
Pellissippi Center
P.O. Box 22991
Knoxville, TN 37933-0991

and

Department of Computer Science
University of Tennessee
Knoxville, TN 37996-1301

Prepared for:

Naval Ocean Research and Development Activity
NSTL, Mississippi 39529-5004

November, 1987

An "expert system" is being developed—

Introduction

This is the final report for the year's work on the project to develop an "expert system" to assist in ~~the~~ analysis and interpretation of mesoscale features in the Atlantic Ocean off the U.S. east coast. ~~NORDA~~ Technical Report 148 ("Development of an Expert System for Interpretation of Oceanographic Images", June, 1986) discusses the approach and gives an English language a statement of a base of knowledge about warm- and cold-core rings (formation, expected movement, evolution and decay) and the Gulf Stream. This knowledge base was formed by discussions with NORDA experts and by review of the technical literature in this rapidly evolving branch of oceanography.

There is not total agreement by oceanographic experts in all aspects of mesoscale features. Updated knowledge and more subtle studies are reported almost every month. (Each entry in the knowledge base in NORDA Technical Report 148 carries a specific literature citation for documentation of the source and date of the information.) The results of a small-sample survey reported in NORDA ~~Technical~~ Note 350 ("Oceanographic Expert System Knowledge Base Evaluation", May, 1987) indicate predominant agreement with the statements in the knowledge base, although there are cases in which the reviewers show sharp division, and there are many cases of "no comment".

It must be emphasized that this system was not designed as a "stand alone" simulation to give accurate, long term predictions; rather, it ~~the~~ system should be regarded as forming hypotheses about the evolution mesoscale features which must then be validated as much as possible by evidence in satellite data. The system as implemented may be stepped by the user for a long time interval, but this does not alter the fact that the system must be viewed as one component in the development of a larger, knowledge-based package for image analysis.

This final report discusses, in order, the items in the Work Statement for Year 1 (Section C.2 in the Solicitation) and the Software Deliverables (Section C.4 in the Solicitation).

Performance of Tasks in the Statement of Work

The following tasks were performed in the work of year 1.

C.2.1 IMPLEMENTATION LANGUAGE

After discussions with NORDA, the language OPS83 was selected for the rule-based coding. This language offers excellent facilities for rule-based programming and has as well a full procedural language capability. It is easily linked to C and FORTRAN, for instance, to make use of a GKS FORTRAN package.

The PASCAL prototype of the original knowledge-based system was recoded in OPS83 and a tape delivered to NORDA. The prototype was a very restricted version of the knowledge base, and even the first delivered OPS83 version was substantially more powerful in that it incorporated more of the knowledge base in its rules.

The approach taken to encoding the rules was to partition the area of the Atlantic Ocean of interest into nine disjoint regions defined by latitude-longitude coordinates. Region-dependent parameters describing expected ring and Gulf Stream activity are read in from data files each time the code executes. This allows "tuning" the system by changing regional parameters in a straightforward way without recompiling.

Subsequently delivered versions in OPS83 included more capability as described below.

C.2.2 EXPAND EXISTING KNOWLEDGE BASE

Updated versions have been supplied to NORDA on tape as significant expansions occurred. In comparison with the prototype and earlier versions, the latest version has more subtle rules, an extension of parts of the knowledge described in NORDA Technical Report 148, certain additional knowledge about mesoscale features obtained from the literature during the year, and some "tuning" of parameters based on studying actual IR images supplied by NORDA during the summer of 1987.

Per agreement with NORDA after discussions of display options in 1987, the current version also provides GKS for a graphical display of the system's inferences about mesoscale features. This required the purchase of a GKS FORTRAN package (Tektronix Plot-10 GKS for DEC VAX systems) and its linking to the OPS83 modules to produce executable files.

Thus, the system EDDIES is a linking of OPS83, C, and FORTRAN modules. FORTRAN is used only because the GKS package is FORTRAN. C is used for mathematical calculations needing the built-in functions of the C math library.

The OPS83 modules and their functions are as follows:

EDDIES.OPS

This is the "start module", i.e., the VAX command RUN EDDIES begins system execution. The module initializes and sets up parameters, then loops during normlal execution through the nine warm-core ring regions and the nine cold-core ring regions, firing the appropriate rules to update rings' status. The module also invokes the procedure to move the Gulf Stream.

EDDYTYPES.OPS

This module contains the declarations of global variables, arrays, records, and working memory elements. It also contains the procedure INIT to initialize critical parameters, to zero critical arrays, and to read into records the values from data files (.DAT files) used to set region parameters.

SETUP.OPS

This module asks the user for Yes or No answers to setup the mode of a given run, e.g., "Use gks (y or n)?" and "Enable debugging details to screen (y or n)?" The module reads in the Gulf Stream boundaries from data files or creates a "nominal Gulf Stream", according to the user's indication of which mode to employ; and it reads in the initial parameters for warm- and cold-core rings from data files.

REGIONS.OPS

This is a small module that determines in which region a ring is located by testing the latitude-longitude of the ring's center with respect to the regions' non-overlapping latitude-longitude limits.

RACYCLE.OPS

This module is the "recognize-act cycle" module which uses built-in functions supplied in OPS83 to determine the rule to be applied to each ring in turn; the module then fires that rule. For example, if more than one ring exists in a given region, this module insures that only one updating rule is applied to each ring in a complete sweep through the regions.

WCRULES.OPS

This module contains the eighteen rules for warm-core rings. Nine of these are called in turn "rule WCR1", "rule WCR2",..., "rule WCR9"; these rules do a basic updating of a ring's status using the nine sets of region parameters, and one of these rules is always applied first to a ring during an updating sweep through the regions. The remaining nine are called "rule WCR1GS" through "rule WCR9GS"; these compute a ring's interaction with the Gulf Stream and adjust the ring's status accordingly. The ring-Gulf Stream interaction is determined by the distance and direction of the center of the ring to the nearest point on the Gulf Stream boundary.

Due to the more extensive computation required, rules "WCR1GS" through "WCR9GS" are more complex than rules "WCR1" through "WCR9". For convenience in documentation and debugging, it was decided to create only two rules per region; but each rule in fact covers many combinations of parameters and computes the appropriate response based thereon.

CCRULES.OPS

This module contains the eighteen rules for cold-core rings similar to those for warm-core rings in the module WCRULES.OPS. The CCR-rules are somewhat more complex than those for warm-core rings because a cold-core ring may have a "looping" motion as it encounters the Gulf Stream, moves away, then reencounters the Gulf Stream many times. Indeed, each rule "CCR1GS" through "CCR9GS" is a quite complex procedure to determine a looping characteristic in its own right. During an updating sweep through the regions, one of the rules "CCR1-9" always fires first for a ring; then the corresponding rule "CCR1GS-9GS" fires to handle the Gulf Stream interaction.

The supporting modules in C are as follows:

MATHE.C

This module contains the procedures to:

(i) compute the distance and direction of the center of a ring to the nearest point on the Gulf Stream boundary,

(ii) create a "nominal Gulf Stream" axis for reference as the nominal center of the Gulf Stream's position, and

(iii) create the "nominal Gulf Stream" boundaries offset 50KM from the nominal axis to give a 100KM-wide reference Gulf Stream when the user indicates that this is to be used instead of Gulf Stream data read from data files.

NRMLZ.C

When the user indicates that Gulf Stream boundary files are to be read in, this module reads the data points from a file UGS.DAT defining the "upper" Gulf Stream boundary and normalizes them to a set of points relative to the "nominal Gulf Stream" axis. The module similarly processes a file LGS.DAT defining the "lower" Gulf Stream boundary. Both the procedure to move the Gulf Stream and the procedure to compute ring-to-Gulf Stream distance require this normalized form.

MOVEGS.C

This module contains the procedure to shift the Gulf Stream data points eastward to simulate Gulf Stream evolution. The procedure reads parameters from a data file MOVEGS.DAT to control its simulation.

GRAPHRTN.C

This module is associated with the Tektronix Plot-10 GKS package. According to the user's response, the module passes the type of workstation into the GKS and establishes the workstation environment. It also contains the procedures to draw the graphics items like the gridlines, the coastline, the Gulf Stream, and the rings.

The single FORTRAN module is required by the GKS package to provide the graphics display:

FO.FOR

This module opens the workstation and passes ID parameters.

The figures at the end of this report demonstrate aspects of the system in operation. Since the purpose of the figures is to show some main points of the running OPS83 code, the figures are simply reproduced by laser line-printer; the actual imagery is, of course, more detailed when suitably displayed on a monitor. The images are NOAA IR supplied by NORDA. In order to prepare figures like these, a version of the code was altered for use with a Perceptics Model 9200 Image Processor with a MicroVAX-II host.

Figure 1 shows a major warm-core ring centered at about 39.5 degrees North, 67 degrees West. Figure 2 is the same region of the Atlantic several weeks later. Figure 3 shows the ring with a display overlay indicating initial working memory entry for the ring. Also labeled are Cape Cod, the approximate 200 meter isobath, and the area of the New England Seamounts for reference.

Figure 4 shows the weekly steps of the knowledge-based system as a sequence of regions of interest in which the system expects to find the evolved warm-core ring. Figure 5 is an expanded subimage of fig. 2 showing the region of interest in which the system expects to find the ring after the passage of several weeks. The light circle overlay highlights the predominant mass of the ring found in ths region and shows it to be inside the circle which was produced by the system.

It must be emphasized that the level of accuracy over seven or eight weeks represented by the above figures is <u>not</u> guaranteed for all rings under all circumstances. The fundamental design principle for the knowledge-based system has always been that it would be expanded for feedback to adjust its working memory entries as it runs. (Coding these expansions were not part of the tasks of the first year.)

<u>C.2.3 EXPAND SCOPE OF SYSTEM</u>

Initial work has been carried out in using low level image processing to look for mesoscale features and in integrating these calculations with the higher level, knowledge-based rules. It has always been intended that the system ultimately be developed to accept "evidence" from processed infrared images and other sources (e.g., altimetry) and to adjust its working memory entries of rings by incorporating this evidence; that is, it is not intended that the system be used as a free-running simulation over long periods of time without additional, corrective inputs.

The most straightforward mechanism under investigation is to have the expert system create a subimage as a region-of-interest in which a ring is expected to be found, then apply image processing methods to that subimage to seek the ring. The methods under study include standard, low level processing like boundary detection as well as innovative methods like Markov random fields. This is illustrated in the figures above.

It should be noted that the actual expansion of the system is a task, C.3.1, scheduled for the second year. Inevitably in so complex a programming job as this, debugging during actual implementation is the only final way to complete the integration of additional computation with the current code.

The approach planned to incorporate other data/information sources is that of multi-sensor fusion in a rule-based system. The requirement here is that the high level, rule-based algorithms be expanded to compute the impact of various data (or evidence) on current knowledge about features; that is, rules in WCRULES.OPS and in CCRULES.OPS must be recoded to include transformation of different evidence to a standard reference so that fusion of evidence may occur. A probability weighting or confidence level parameter would be included to assist in the data fusion. The plan is to have certain parameters controlling this fusion read from a data file so that tuning can be accomplished without recompiling.

As with recognition of mesoscale features, multi-sensor fusion must be coded, tested, and debugged in order to expand the current system to this new capability.

C.2.4. DEVELOP PLANS FOR EXTENDING APPLICATION

An initial literature search was begun to seek references in the technical literature to detection and delineation of sea ice. Since this detection and delineation is to use multiple sources of sensor data, the plan has been to use the basic format of multi-sensor fusion for mesoscale features, as coded and refined first. This would result in a system that handles multiple sources of data in a consistent way.

Software Deliverables

C.4.1. Throughout the year, the updated versions of the OPS83, C, and FORTRAN code have been provided to NORDA. The most recent code was provided in October 1987.

A printed listing of all code has been provided.

Section C.2.2 above explains the functions of the modules. After entering RUN EDDIES, the user is presented with a series of simple questions with Yes or No answers to establish the mode of a particular run.

C.4.2. All variables in OPS83 are typed. In-line comments are provided to explain each variable's use; for example, the global variables are explained in the module EDDYTYPES.OPS.

Figure 1. Region with warm-core ring

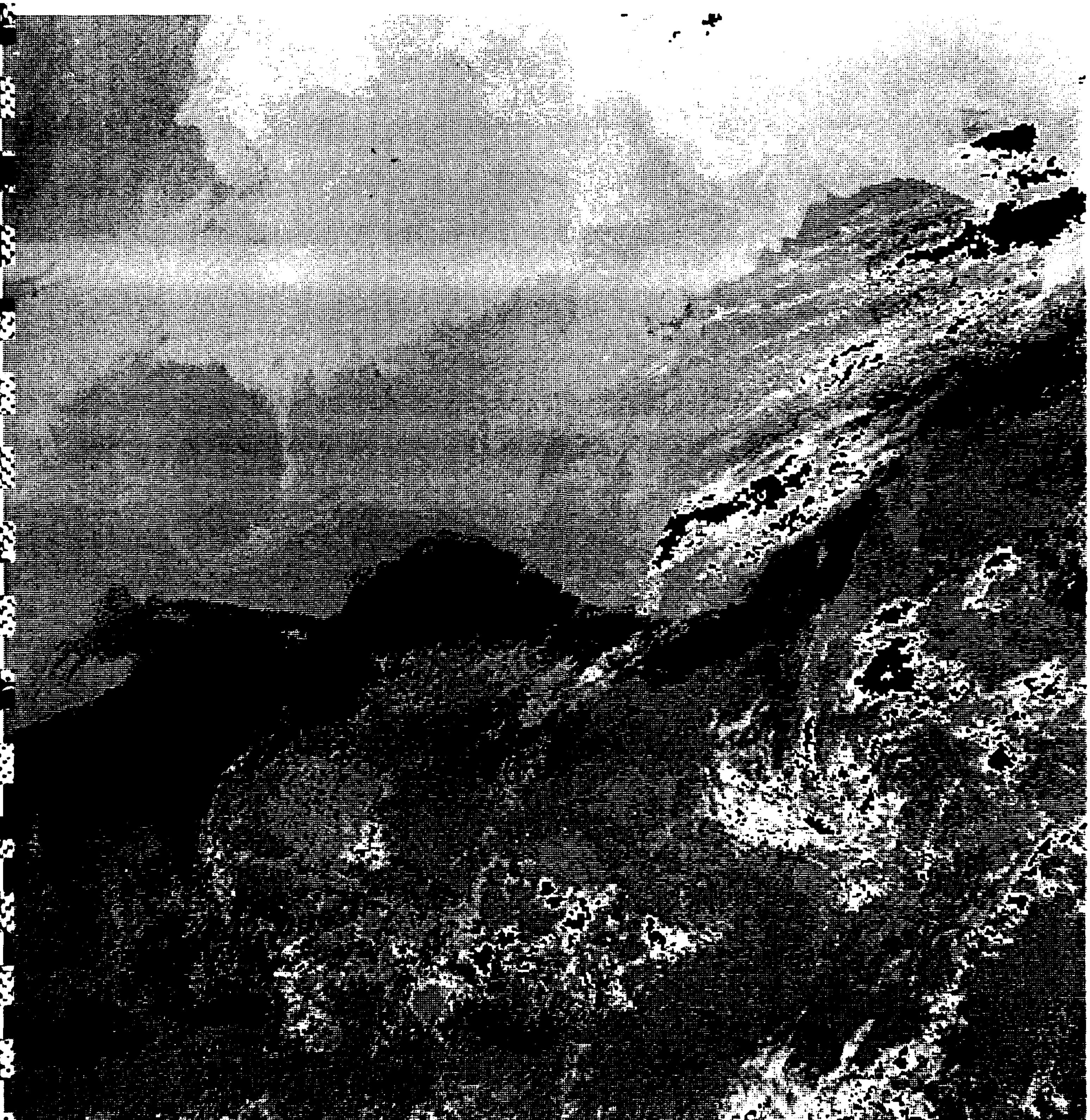


Figure 2. Same region at a later date
(approximately two months).

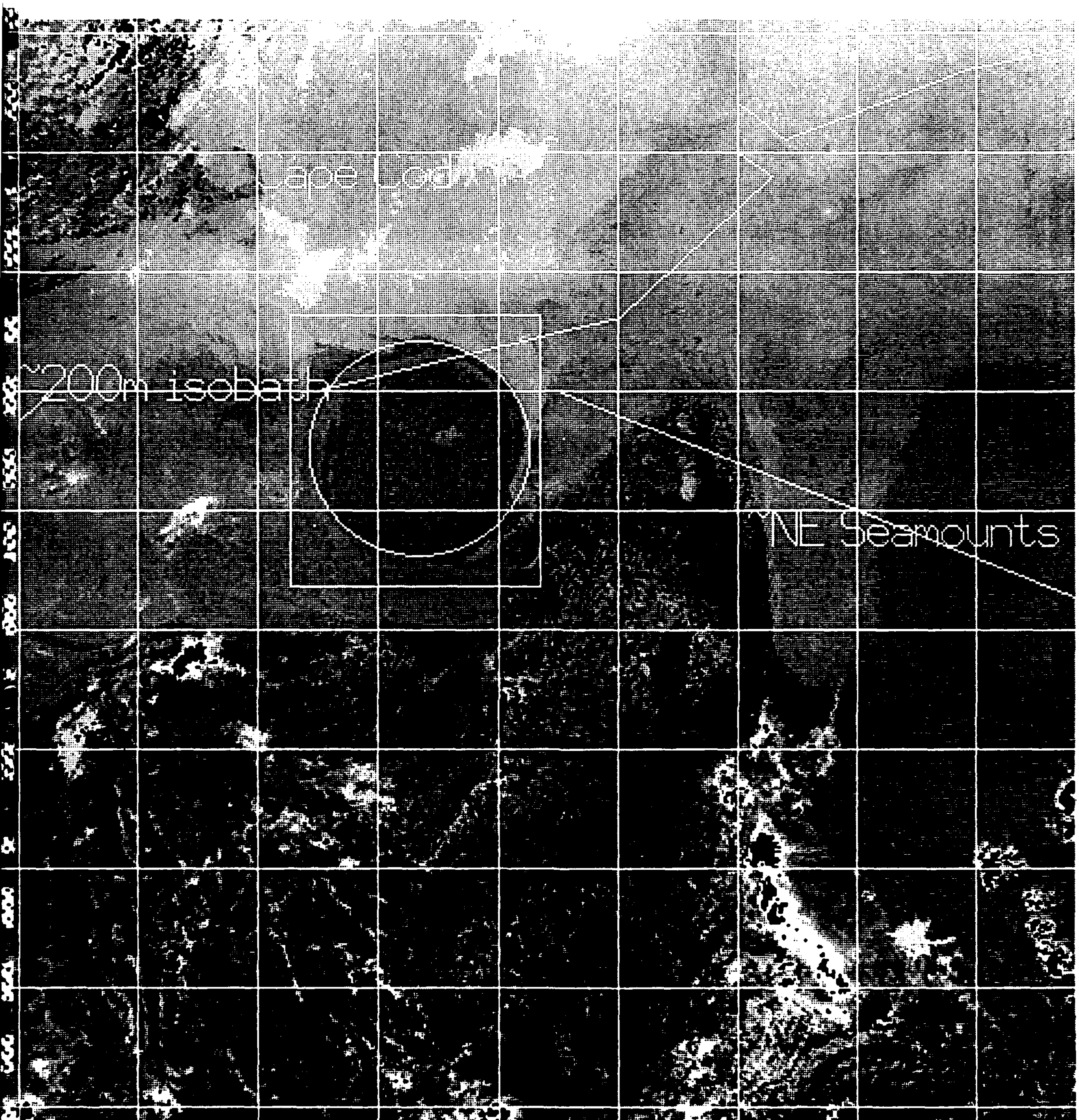


Figure 3. Graphics overlay on
image of figure 1.

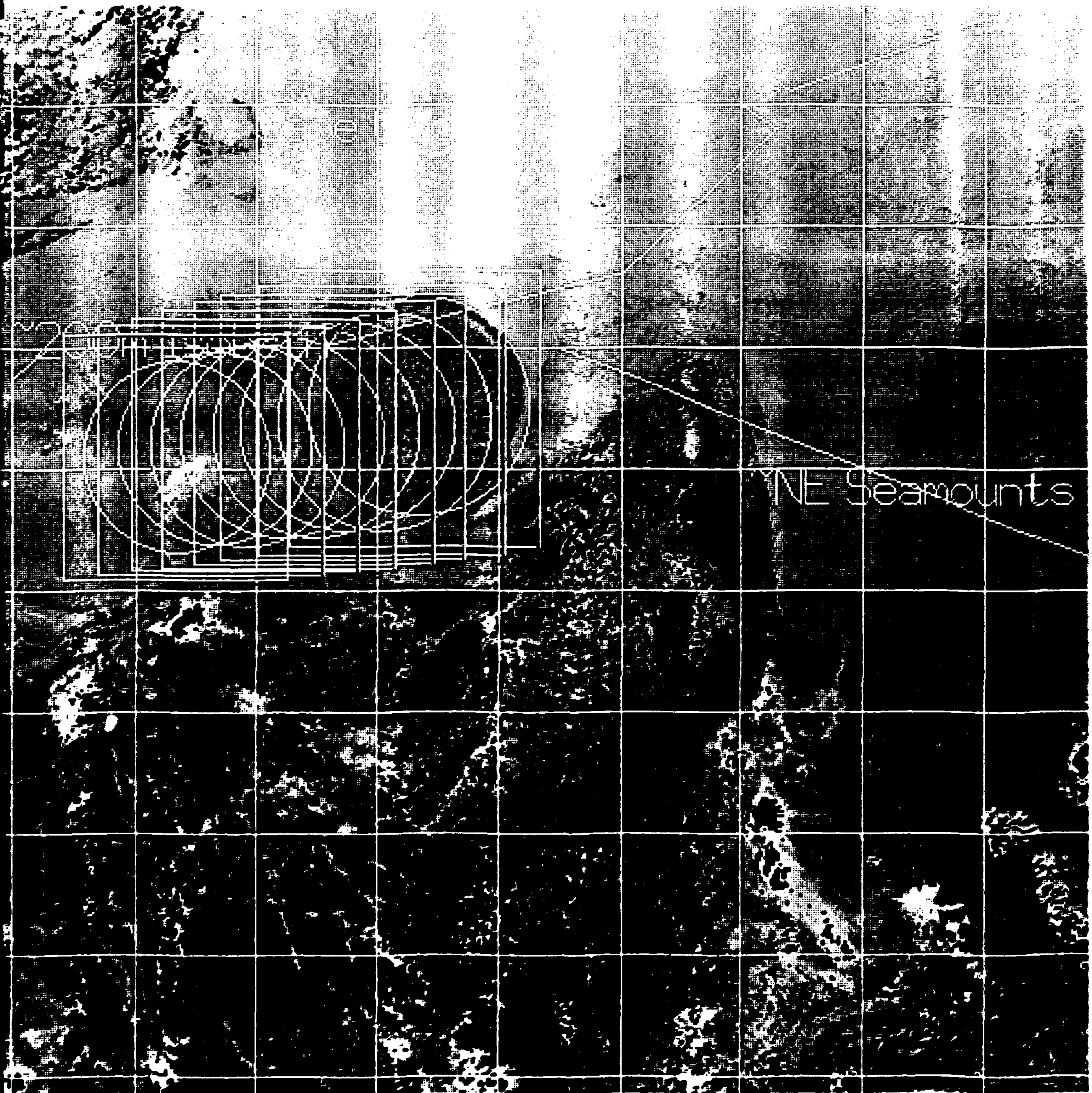
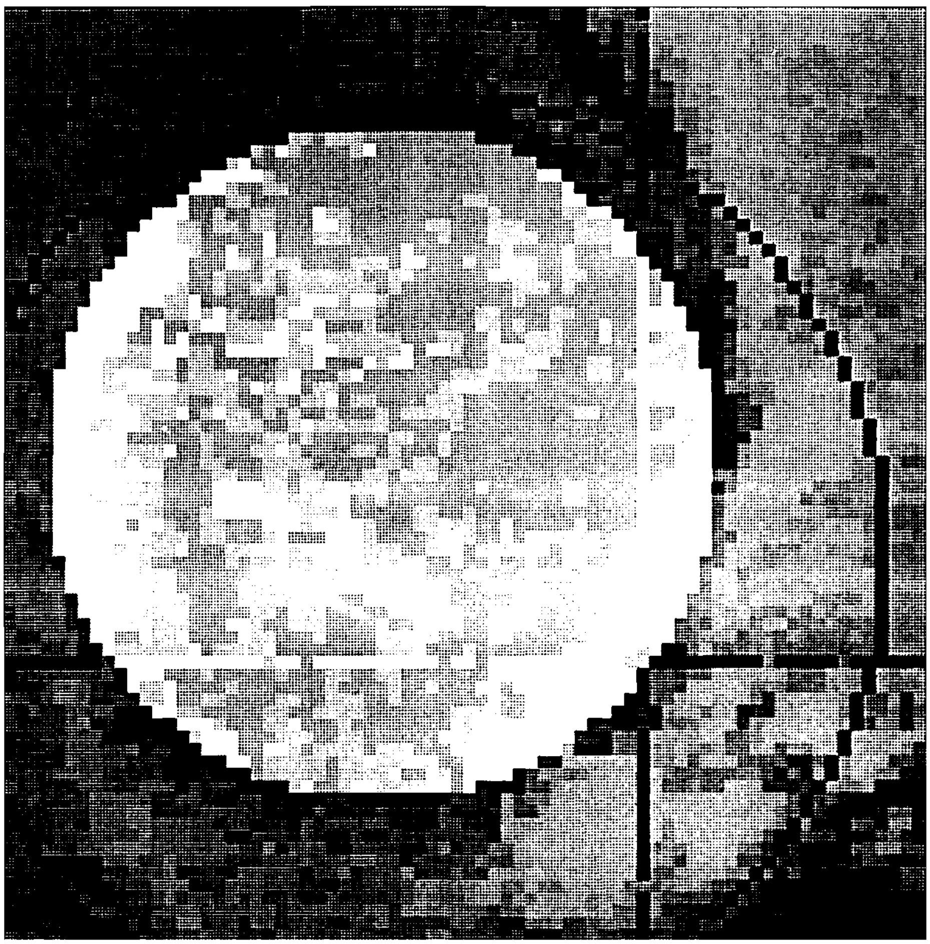


Figure 4. Sequence of regions of interest
generated by OPS83 code.



END
DATE
FILMED
MARCH
1988
DTIC